

PROG5 L3 - Projet, année 2016-2017

Realisation d'un editeur de liens | Phase de Fusion

GROUPE 3 : BLEUZE BAPTISTE, BOCHATON MATHIEU, DA SILVA REMI, LAFONT ANTHONY, THISSE NOEMI

1) Mode d'emploi

COMPILATION

via le MAKEFILE fourni

LANCEMENT DU PROJET

`./fusionL3G3 [--options]`

OPTIONS

- `[-1<nom_fichier>]` : met en mémoire le contenu du fichier objet translatable `<nom_fichier>` (information requise) (chemin relatif) si présent
- `[-2<nom_fichier>]` : met en mémoire le contenu du fichier objet translatable `<nom_fichier>` (information requise) (chemin relatif) si présent
- `[-H]` : affiche les options du programme fusionL3G3 (usage)
- `[-d]` : active le flag debug
- `[-h]` : affiche l'entête du fichier 1 (si présent) (cf phase 1 étape 1)
- `[-S]` : affiche la table des sections du fichier 1 (si présent) (cf phase 1 étape 2)
- `[-x<number_or_name>]` : affiche le contenu d'une section du fichier 1 (si présent) `number_or_name` (information requise) peut être l'index ou le nom de section. (cf phase 1 étape 3)
- `[-X]` : affiche le contenu de toutes les sections du fichier 1 (si présent)
- `[-s]` : affiche la table des symboles du fichier 1 (si présent) (cf phase 1 étape 4)
- `[-r]` : affiche la table des relocations du fichier 1 (si présent) (cf phase 1 étape 5)
- `[-f]` : réalise la fusion entre les fichiers 1 et 2 (si présents) (cf phase 2 étape 6-8)
- `[-c<nom_fichier>]` : reconstruit le fichier 1 (si présent) (cf phase 2 étape 9) au format ELF sous le nom `nom_fichier` (produit à la racine)

FONCTIONNALITÉS :

- affiche des informations à propos d'un fichier au format ELF (`-1nom1`) selon l'option demandée sur le modèle de celle de `readelf` (`-h`, `-S`, `-xnom_section`, `-X`, `-s`, `-r`). Le programme n'accepte qu'une option avant de s'arrêter.
- réalise la fusion des deux fichiers (`-1nom1`, `-2nom2`). Le résultat est reconstruit dans un fichier dont le nom est celui passé en paramètre (écrase tout fichier existant).

2) Structure du code

Pour faciliter le code, nous avons créé une structure ELF pour stocker toutes les données d'un fichier ELF (entête du fichier, entête des sections et contenu des sections) ainsi que d'autres informations pratiques pour manipuler notre structure (indices, table des symboles, etc...). Cette structure a été repensée à chaque étape du projet pour faciliter à la fois la lecture (phase 1) et la fusion (phase 2). Elle est déclarée dans le fichier type.h

Chaque étape du projet a été développée dans des fichiers distincts sous forme d'une bibliothèque de fonctions réutilisable dans notre main :

- "lecture_fichier_ELF.c" réalise la mise en mémoire/structure d'un fichier donné en paramètre pour peu qu'il soit au format ELF. Reconstitue également un fichier au format ELF si besoin.
- "affichage.h" fait le lien entre toutes nos bibliothèques gérant les contenu à afficher
- "afficher_XXX_ELF" gère les étapes 1 à 5 de la phase 1
- "tableau.h" définit les macros nécessaires à notre affichage
- "fusion_progbits_ELF" gère l'étape 6, c'est-à-dire fusionne les sections de type PROGBITS des deux fichiers donnés en paramètre, et les modifications que cela engendre dans le nouveau fichier (décalage ou ajout de contenu)
- "fusion_symboles_ELF" gère l'étape 7, gère la fusion des tables de symboles nécessaires après l'étape 6.

3) Fonctionnalités

Phase 1

L'ensemble de la phase une est fonctionnelle, les affichages suivants sont fonctionnels :

1. L'entête du fichier ELF
2. La table des en-têtes de sections
3. Le contenu d'une section en particulier
4. La table des symboles
5. La table des réimplantations

Un soin tout particulier a été porté au rendu de l'affichage pour un confort d'utilisation différent de celle de la commande readelf. Les informations données sont sensiblement les mêmes, cependant un diff entre les deux affichages produits n'a aucune valeur.

Phase 2

Pour la phase deux, nous recréons une structure ELF3 depuis le premier fichier fourni et propageons les modifications nécessaires:

6. La fusion des sections PROGBITS de deux fichiers ELF est fonctionnelle.
7. La fusion des tables des symboles est fonctionnelle.
8. La fusion des tables de réimplantation n'est pas implémentée.
9. La création d'un fichier .o à partir d'une structure ELF est implémentée.

4) Bogues connus mais non résolus

Dans l'état actuel des tests aucun bug connu n'est resté irrésolu.

5) Tests effectués

Chaque étape de la phase 1 a été vérifiée, au fur et à mesure du développement à partir d'un jeu de tests non exhaustif . Nous n'avons pas mis en place de jeu de tests automatisé sur un panel important de fichiers objet translatable.

6) Journal de répartition des tâches

	mardi 03/01	mercredi 04/01	jeudi 05/01	vendredi 06/01	lundi 09/01	mardi 10/01	mercredi 11/01	jeudi 12/01
Baptiste	Lecture du sujet, de la documentation ELF, mise en place d'un git, étape 1	étape 1	factorisation du code, amélioration du code produit à chaque étape					étape 7
Rémi		étape 4			factorisation du code, etc..		étape 7	relecture code + préparation démon
Noémi		étape 2	étape 3	étape 5	étape 6	étape 6	amélioration du main	
Anthony			étape 5			étape 9	étape 9	
Mathieu								